CLAIMS

What is claimed is:

1. A computerized method for sequencing and reassembling messages from protocol data units exchanged in a communications channel between two computers, the method comprising:

creating a protocol flow object to represent each protocol layer used by the communications channel, each protocol flow object having a circuit element associated with each transmission direction in the channel;

arranging the protocol flow objects in a logical tree structure that mirrors a hierarchy for the protocol layers;

creating circuit flow objects for each protocol layer to represent the protocol data units for the protocol layer immediately higher in the hierarchy;

associating a transmission direction with each circuit flow object;

linking each circuit flow object for a protocol layer to the circuit element of the representative protocol flow object that matches the transmission direction associated with the circuit flow object;

sequencing the circuit flow objects linked to a particular protocol flow object when specified by the protocol layer represented by the particular protocol flow object; and

reassembling the messages from the circuit flow objects linked to the protocol flow object at the top of the tree structure.

2. The method of claim 1, wherein creating the circuit flow objects for each protocol layer comprises:

creating the circuit flow objects for the protocol flow object at the bottom of the tree structure by extracting data from the protocol data units for the protocol layer lowest in the hierarchy; and

creating the circuit flow objects for the remaining protocol flow objects in the tree structure by extracting data from the circuit flow objects linked to the protocol flow object immediately lower in the tree structure.

3. The method of claim 1, wherein a circuit flow object comprises a vector list to represent fragmented data.

4. The method of claim 3, wherein a vector list comprises a vector specifying a protocol data unit number, a length value, and an offset value for each fragment of the fragmented data.

5. The method of claim 4 further comprising:

reassembling the fragmented data in accordance with the vectors in a vector list.

6. The method of claim 1 wherein the protocol flow objects are created in order from the bottom to the top of the hierarchy.

7. The method of claim 6, wherein the circuit flow objects for a current protocol flow object are created before creating the protocol flow object for the protocol layer immediately above the current protocol flow object in the hierarchy.

8. The method of claim 1 wherein arranging the protocol flow objects into a logical tree structure comprises:

creating multiple branches in the tree structure when a plurality of protocol layers are immediately above a current protocol layer in the hierarchy, each branch corresponding to one of the plurality of protocol layers.

9. The method of claim 1 further comprising:

determining the protocol layers in the hierarchy.

10. The method of claim 1 further comprising:

storing the protocol flow objects and the circuit flow objects in a flow object database.

11. A computer-readable medium having computer-executable instructions to a cause a computer to perform a method comprising:

creating a protocol flow object to represent each protocol layer used by a communications channel, each protocol flow object having a circuit element associated with a transmission direction in the channel;

arranging the protocol flow objects in a logical tree structure that mirrors a hierarchy for the protocol layers;

creating circuit flow objects for each protocol layer to represent the protocol data units for the protocol layer immediately higher in the hierarchy;

associating a transmission direction with each circuit flow object;

linking each circuit flow object for a protocol layer to the circuit element of the representative protocol flow object that matches the transmission direction associated with the circuit flow object;

sequencing the circuit flow objects linked to a particular protocol flow object when specified by the protocol layer represented by the particular protocol flow object; and

reassembling the messages from the circuit flow objects linked to the protocol flow object at the top of the tree structure.

12. The computer-readable medium of claim 11 having further computer-executable instructions comprising:

creating the circuit flow object for the protocol flow object at the bottom of the tree structure by extracting data from the protocol data units for the protocol layer lowest in the hierarchy; and

creating the circuit flow objects for the remaining protocol layers by extracting data from the circuit flow objects linked to the protocol flow object immediately lower in the tree structure.

13. The computer-readable medium of claim 11 having further computer-executable instructions comprising:

creating a circuit flow object as a vector list to represent fragmented data.

14. The computer-readable medium of claim 13 having further computer-executable instructions comprising:

creating a vector list from a plurality of vectors, each vector specifying a protocol data unit number, a length value, and an offset value for a fragment of the fragmented data.

15. The computer-readable medium of claim 14 having further computer-executable instructions comprising:

reassembling the fragmented data in accordance with the vectors in a vector list.

16. The computer-readable medium of claim 11 having further computer-executable instructions comprising:

creating multiple branches in the tree structure when a plurality of protocol layers are immediately above a current protocol layer in the hierarchy, each branch corresponding to one of the plurality of protocol layers.

17. The computer-readable medium of claim 11 having further computer-executable instructions comprising:

determining the protocol layers in the hierarchy.

18. The computer-readable medium of claim 11 having further computer-readable instructions comprising:

storing the protocol flow objects and the circuit flow objects in a flow object database.

19. A computer-readable medium having stored thereon an protocol flow object data structure comprising:

a key field containing data representing an identifier for a connection between two computers at a protocol layer;

a primary circuit element containing data representing a link to a series of protocol data units flowing in one direction in the connection identified by the key field; and

an alternate circuit element containing data representing a link to a series of protocol data units flowing in an opposite direction in the connection identified by the key field.

20. The computer-readable medium of claim 19, wherein the links comprise hash tables for identifying the series of data units.

21. A computer-readable medium having stored thereon a re-assembly vector comprising:

a protocol data unit field containing data representing a number for a protocol data unit;

a length field containing data representing a length of a data payload in the protocol data unit identified by the protocol data unit field; and

an offset field containing data representing a starting position of the data payload in the protocol data unit identified by the protocol data unit field.

22. A computer-readable medium having stored thereon a flow object data structure comprising:

a plurality of protocol flow objects, each protocol flow object comprising:

a key field containing data representing an identifier for a connection between two computers at a protocol layer;

a primary circuit element containing data representing a link to a series of protocol data units flowing in one direction in the connection identified by the key field; and

an alternate circuit element containing data representing a link to a series of protocol data units flowing in an opposite direction in the connection identified by the key field;

a tree structure comprising a plurality of entries, each entry comprising:

a protocol field containing data representing the identifier for one of the plurality of protocol flow objects;

a lower protocol field containing data representing the identifier for the protocol flow object immediately lower in a protocol layer hierarchy relative to the protocol flow object identified by the protocol field; and

a higher protocol field containing data representing the identifier for the protocol flow object immediately higher in the protocol layer hierarchy relative to the protocol flow object identified by the protocol field.

23. The computer-readable medium of claim 22 further comprising:

a plurality of circuit flow objects, each circuit flow object containing data representing one of the protocol data units.

24. A computerized system comprising:

a processor;

a memory coupled to the processor through a bus;

a computer-readable medium coupled to the processor through the bus;

a plurality of protocol interpreters stored on the computer-readable medium for execution by the processor; and

a decode engine executed from the computer-readable medium to cause the processor to

create protocol flow objects representing protocol layers and circuit flow objects representing data flows at the protocol layers,

extract data from the circuit flow objects representing protocol data units at a particular protocol layer as directed by one of the protocol interpreters,

sequence the circuit flow objects representing the protocol data units at a particular protocol layer if directed by one of the protocol interpreters, and

reassemble messages from the circuit flow objects representing the protocol data units at a particular protocol layer if directed by one of the protocol interpreters.

25. The computer system of claim 24, wherein the decode engine further causes the processor to store the protocol flow objects and circuit flow objects in a flow database, logically link the protocol flow objects into a hierarchical tree structure, and to logically link the circuit flow objects to the protocol flow objects.

26. The computer system of claim 24, wherein the decode engine further causes the processor to create a circuit flow object as a vector list to represent fragmented data.

27. The computer system of claim 26, wherein the decode engine further causes the processor to create a vector list from a plurality of vectors, each vector specifying a protocol data unit number, a length value, and an offset value for a fragment of the fragmented data.

28. The computer system of claim 27, wherein the decode engine further causes the processor to reassemble the fragmented data in accordance with the vectors in a vector list.

29. A method of communicating between a protocol interpreter and a segmentation and re-assembly decode engine for a communications network comprising:

issuing, by the protocol interpreter, an add data unit command;

receiving, by the segmentation and re-assembly decode engine, the add data unit command; and

issuing, by the segmentation and re-assembly decode engine in response to receiving the add data unit command, an instruction to a flow object data base.

30. The method of claim 29, wherein the instruction is selected from the group consisting of adding a circuit flow object to the flow object data base, associating a circuit flow object to a protocol flow object in the flow object data base, retrieving a circuit flow object from the flow object data base, and sequencing a circuit flow object relative to other circuit flow objects in the data base.